An Empirical Study on Distribution Shift Robustness From the Perspective of Pre-Training and Data Augmentation

Ziquan Liu[1]*, **Yi Xu**[2✉], **Yuanhong Xu**[3], **Qi Qian**[3], **Hao Li**[3], **Rong Jin**[3], **Xiangyang Ji**[4], **Antoni B. Chan**[1]

[1]Department of Computer Science, City University of Hong Kong
[2]School of Artificial Intelligence, Dalian University of Technology
[3]DAMO Academy, Alibaba Group
[4]Department of Automation, Tsinghua University
ziquanliu2-c@my.cityu.edu.hk, yxu@dlut.edu.cn, {yuanhong.xuyh, qi.qian, lihao.lh}@alibaba-inc.com, rongjinemail@gmail.com, xyji@tsinghua.edu.cn, abchan@cityu.edu.hk

Abstract

The performance of machine learning models under distribution shift has been the focus of the community in recent years. Most of current methods have been proposed to improve the robustness to distribution shift from the algorithmic perspective, i.e., designing better training algorithms to help the generalization in shifted test distributions. This paper studies the distribution shift problem from the perspective of pre-training and data augmentation, two important factors in the practice of deep learning that have not been systematically investigated by existing work. By evaluating seven pre-trained models, including ResNets [1] and ViT's [2] with self-supervision and supervision mode, on five important distribution-shift datasets, from WILDS [3] and DomainBed [4] benchmarks, with five different learning algorithms, we provide the first comprehensive empirical study focusing on pre-training and data augmentation. With our empirical result obtained from 1,330 models, we provide the following main observations: 1) ERM combined with data augmentation can achieve state-of-the-art performance if we choose a proper pre-trained model respecting the data property; 2) specialized algorithms further improve the robustness on top of ERM when handling a specific type of distribution shift, e.g., GroupDRO [5] for spurious correlation and CORAL [6] for large-scale out-of-distribution data; 3) Comparing different pre-training modes, architectures and data sizes, we provide novel observations about pre-training on distribution shift, which sheds light on designing or selecting pre-training strategy for different kinds of distribution shifts. In summary, our empirical study provides a comprehensive baseline for a wide range of pre-training models fine-tuned with data augmentation, which potentially inspires research exploiting the power of pre-training and data augmentation in the future of distribution shift study.

1 Introduction

Machine learning (ML) has received much success in many computer vision applications such as image classification. However, it heavily depends on an in-distribution (ID) assumption that the training data and testing data are identically and independently from the same data distribution. Unfortunately, this ID assumption may be hardly satisfied in practice, which leads to distribution

*Work partially done during an internship at DAMO Academy, Alibaba Group. ✉ indicates corresponding author.

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

	WILDS-Waterbirds		WILDS-FMoW		WILDS-Camelyon	WILDS-iWildCam		DomainNet
Metric	WG Acc.	Avg. Acc.	WG Acc.	Avg. Acc.	OOD Acc.	OOD Macro F1	ID Macro F1	Avg. Acc.
SotA	91.4(1.1) [5]	93.5(0.3) [5]	35.5(0.8) [16]	52.8(1.2) [16]	93.3(1.0) [17]	38.5(0.6) [18]	52.8(1.4) [18]	49.8(0.1) [19]
Our Best	**92.6(0.8)**	**94.1(0.7)**	**40.7(1.0)**	**57.4(2.1)**	**94.7(0.2)**	**43.2(0.9)**	52.1(2.4)	**52.1(1.1)**
Our ERM Best	**92.6(0.8)**	**94.1(0.7)**	40.2(1.6)	57.1(1.0)	94.5(0.5)	41.3(2.4)	**55.7(2.1)**	49.8(0.1)

Table 1: The datasets with distribution shift evaluated in this paper. Spurious correlation (Waterbirds), subpopulation shift (FMoW) and out-of-distribution generalization (Camelyon, iWildCam and DomainNet) are considered. Our best result achieved by ERM combined with data augmentation matches state-of-the-art results on the five datasets, demonstrating the importance of selecting pre-trained models in improving distribution shift robustness. The state-of-the-art result is from [20] and [21] for a single model without model averaging and ensemble.

shifts. In the past several years, there has been a line of work proposed to study distribution shifts from different perspectives [7, 5, 8, 6]. Arjovsky et al. [7] proposed Invariant Risk Minimization (IRM) to learn invariant prediction across multiple training environments, which does not need the assumption of i.i.d. in training and testing. Follow-up works of IRM consider both theoretical and empirical aspects (e.g., see [9, 10, 11, 12]). On the other hand, robust optimization is a popular technique for distribution shift problems, which minimizes the worst-case loss. Examples include Distributionally Robust Optimization (DRO) [13, 14, 5], minimax Risk Extrapolation (MM-REx) [8], and Heterogeneous Risk Minimization (HRM) [15]. However, most existing methods focus on developing new specialized learning algorithms to improve the generalization under distribution shifts, without considering deep learning (DL) characteristics such as training procedure and model architecture which are the important components to performance improvement in deep learning.

In the paper, we empirically investigate the importance of pre-training and model architectures for generalization under distribution shift, with a focus on image classification tasks. To this end, we consider Empirical Risk Minimization (ERM) [22], the most commonly-used learning paradigm in DL. We identify the source of naturally arising distribution shifts and study the impact of pre-training, which is carried out on a well-controlled training set without distribution shifts such as ImageNet [23], on downstream datasets with distribution shifts when fine-tuning with data augmentations. The contributions of our paper are:

1. We provide the first extensive empirical study on how pre-training and data augmentation affect robustness to distribution shifts on various computer vision datasets. We consider seven pre-training models (including self-supervised and supervised ResNets and ViT's), three data augmentations (including a group-aware mixup), and five datasets with distribution shifts (e.g., spurious correlation, subpopulation shift and OOD generalization), totaling 1,330 trained models.
2. We find that specialized algorithms such as GroupDRO [5] and CORAL [6] for distribution shifts achieve the state-of-the-art performance with proper pre-trained models on some datasets, while ERM with data augmentation is also a quite competitive baseline.
3. Comparing different pre-training strategies, models and data sizes, we summarize key observations about the impact of pre-training on distribution shifts, and provide practical tips on selecting pre-trained models under different types of distribution shifts.

2 Related Work

Out-of-Distribution Generalization. The OOD generalization task assumes that the generation processes of training and test samples conditioned on the target label are different. For example, the environment or background of objects changes between training and test (WILDS-Camelyon, WILDS-iWildCam, ImageNet-C [24]) or the visual features of objects are changed during test (DomainNet, ImageNet-A [25]). Various methods have been proposed to improve the generalization under OOD, including domain adversarial learning [26, 27] and domain feature aligning [6, 28]. We propose to study the performance of different OOD generalization algorithms including the popular method CORAL [26], when different pre-trained models and data augmentation tricks are used. Similar to the finding for spurious correlation, different pre-trained models have substantially different impacts on OOD generalization. Key observations and tips about OOD generalization are provided based on our empirical results. Please refer to the survey [29] for more methods on OOD generalization.

The Impact of Pre-Training and Data Augmentation. [30] summarizes a framework for distribution shift, consisting of spurious correlation, low-data shift and unseen data, and evaluates the performance of representation learning, data augmentation and neural architectures. However, [30] analyzed those

			ERM			GroupDRO	CORAL
PT mode	Model	PT data	Data Aug.	Mixup	GroupMix		
MoCo	ViT	IN-1k	84.7(1.3)	83.5(2.8)	85.8(1.4)	**86.7(0.8)**	83.3(1.9)
MoCo	R50	IN-1k	82.5(0.8)	86.7(0.7)	87.7(0.7)	**88.1(0.8)**	83.2(1.8)
MAE	ViT	IN-1k	81.7(2.0)	80.3(1.5)	82.2(2.5)	**87.4(1.0)**	80.0(2.3)
Sup.	ViT	IN-1k	76.3(2.4)	79.5(4.5)	84.1(1.9)	**87.0(1.1)**	81.4(1.7)
	R50	IN-1k	80.7(1.1)	82.6(2.5)	85.0(2.3)	**87.6(0.4)**	79.4(1.9)
	ViT	IN-21k	88.5(0.6)	88.2(1.2)	91.5(0.8)	**92.6(0.5)**	86.5(1.3)
	R50	IN-21k	82.9(1.4)	86.5(1.3)	**87.5(1.6)**	86.6(1.1)	83.2(1.2)
Avg. Over Models			82.5	83.9	86.3	**88.0**	82.4

Table 2: WILDS-Waterbirds Result. We report the worse-group accuracy. The bold numbers are the best performance in the row and the green numbers are the best performance in a column. GroupDRO is a strong algorithm for spurious correlation and GroupMix substantially improves the DA and Mixup baseline.

			ERM			GroupDRO	CORAL
PT mode	Model	PT data	Data Aug.	Mixup	GroupMix		
MoCo	ViT	IN-1k	36.2(1.4)	**37.0(1.7)**	36.5(1.5)	35.2(1.6)	35.8(1.7)
MoCo	R50	IN-1k	36.4(2.3)	36.0(1.2)	**37.1(1.4)**	37.2(1.1)	36.1(0.7)
MAE	ViT	IN-1k	39.1(0.5)	38.8(1.1)	40.2(1.6)	37.9(1.4)	**40.4(0.8)**
Sup.	ViT	IN-1k	34.5(0.8)	34.4(0.3)	35.8(0.7)	35.8(1.8)	**35.9(1.2)**
	R50	IN-1k	34.9(1.9)	35.9(1.8)	**36.8(1.3)**	36.1(1.6)	34.2(1.2)
	ViT	IN-21k	37.9(1.3)	38.5(0.9)	38.6(1.5)	39.0(1.0)	**40.7(1.0)**
	R50	IN-21k	37.2(2.9)	38.4(0.6)	**39.0(0.8)**	37.1(1.9)	36.6(0.8)
Avg. Over Models			36.6	37.0	**37.7**	36.9	37.1

Table 3: WILDS-FMoW Result. The worse-group accuracy is reported. ERM is a quite competitive baseline and the self-supervised pre-trained model MAE is more suitable to the satellite imaging data than other models.

factors *independently* and only evaluated the pre-trained ResNet50. In contrast, our paper investigates the de facto training setting in distribution shift problem, i.e., fine-tuning a pre-trained model on a target task, and we comprehensively evaluate the performances of different pre-training models and neural architectures when combined with data augmentation and fine-tuning. [4] focuses on the role of data augmentation on domain generalization, where the visual features of objects are shifted during test such as DomainNet [31] (Fig. 1), and draws a similar conclusion that ERM is a strong baseline in their context. Our work uses the largest dataset in [4] as representative of domain shift and studies the impact of pre-training in addition to the data augmentation. [32] provides a nice benchmark on distribution shift consisting of different levels of semantic hierarchies with ImageNet [23]. We do not evaluate on [32] because our study focuses on the performance of pre-trained models on ImageNet in various downstream applications, instead of the subpopulation shift in ImageNet data. [33] studies the domain generalization from the perspective of pre-training. The major differences are: 1) our paper has a focus on both pre-trained models and data augmentation, especially mixup and the proposed GroupMix, while [33] only considers pre-trained models; 2) we evaluate not only supervised pre-trained models but also self-supervised pre-trained ViT and ResNet models, including both MoCo and MAE training, while [33] only considers (semi) supervised pre-trained models; 3) our paper evaluates on the WILDS benchmark in addition to the domain generalization data. Similar to our GroupMix, [16] proposes to use a selective mix-up augmentation to interpolate samples with the same label or the same environment. The difference is that we investigate the performance of GroupMix on a variety of pre-trained models to reveal the impact of pre-training on the GroupMix. [34] also studies the impact of pre-training on robustness to distribution shift, where they assume there is an OOD training set for fine-tuning the classification layer. In our paper, there is no such a strong assumption and we not only consider pre-training but also data augmentation in the fine-tuning.

3 Experiment Settings and Results

Our experiment uses seven pre-trained models, i.e., MoCo-ViT, MoCo-R50 [35], MAE pre-trained [36] on ImageNet-1k (IN-1k), Supervised ViT and R50 pre-trained on ImageNet-1k and ImageNet-21k (IN-21k). ERM, Mixup, GroupDRO, group-based Mixup and CORAL are evaluated on four WILDS datasets (Waterbirds [5], Camelyon [37], FMoW [38], iWildCam [39]) and the large-scale domain generalization dataset DomainNet [31]. See the Appendix for details of experiment.

We next report the experimental results. Data augmentation (DA), MixUp and GroupMix are included as the *generalized* ERM algorithm since they are often considered as general data augmentation methods in DL and simple to implement. Thus, we denote the DA, mixup and GroupMix as *general*

PT mode	Model	PT data	ERM Data Aug.	 Mixup	 GroupMix	GroupDRO	CORAL
MoCo	ViT	IN-1k	89.9(2.1)	92.6(0.7)	92.6(0.9)	89.7(2.0)	**92.7(0.4)**
MoCo	R50	IN-1k	90.9(1.3)	**91.8(1.8)**	90.6(2.9)	91.2(1.6)	88.2(4.6)
MAE	ViT	IN-1k	93.7(0.7)	94.4(0.4)	94.5(0.5)	94.4(0.2)	**94.7(0.2)**
Sup.	ViT	IN-1k	90.1(1.2)	92.2(1.8)	93.3(0.8)	**93.3(0.5)**	92.9(1.1)
	R50	IN-1k	85.5(4.6)	78.9(11.6)	80.3(8.5)	**87.5(3.9)**	78.9(11.6)
	ViT	IN-21k	93.9(0.7)	93.5(0.7)	**94.2(0.3)**	90.8(2.3)	93.5(0.7)
	R50	IN-21k	**93.7(0.3)**	91.1(1.1)	89.6(2.7)	93.1(1.7)	91.1(1.1)
Avg. Over Models			91.1	90.6	90.7	**91.4**	90.3

Table 4: WILDS-Camelyon Result. We report the OOD average accuracy. GroupMix and CORAL perform the best on Camelyon and MAE is the best pre-trained model for the tissue slide data in most cases.

PT mode	Model	PT data	ERM Data Aug.	 Mixup	 GroupMix	GroupDRO	CORAL
MoCo	ViT	IN-1k	34.0(0.5)	31.7(2.1)	32.8(0.7)	13.5(1.0)	**35.2(0.6)**
MoCo	R50	IN-1k	34.5(1.5)	36.8(1.6)	36.0(1.1)	20.8(0.8)	**37.2(1.1)**
MAE	ViT	IN-1k	28.4(2.0)	27.3(2.2)	28.0(2.9)	9.2(0.9)	**31.4(2.6)**
Sup.	ViT	IN-1k	38.9(0.7)	41.0(0.7)	40.9(1.2)	20.5(0.5)	**41.9(0.9)**
	R50	IN-1k	32.2(1.2)	32.8(1.4)	31.9(0.9)	18.9(0.8)	**33.4(0.5)**
	ViT	IN-21k	39.0(2.7)	**41.3(2.4)**	41.1(1.4)	19.0(1.2)	36.4(1.6)
	R50	IN-21k	40.9(1.3)	40.8(0.4)	41.2(1.3)	24.0(0.8)	**43.2(0.9)**
Avg. Over Models			35.4	36.0	36.0	18.0	**37.0**

Table 5: WILDS-iWildCam Result. The macro F1 score of OOD data is reported as in WILDS benchmark. CORAL is the best algorithm in most cases while GroupMix is also a strong baseline. In the wild recognition task, supervised pre-training is generally better than the self-supervised counterpart.

DA in the following content. In our experiment, DA is used with ERM (mixup and GroupMix), GroupDRO and CORAL to make a fair comparison. On four WILDS datasets and DomainNet, we report the result of fine-tuning seven pre-trained models with three different algorithms in Tabs. 2-6. The bold numbers indicate the algorithm with the best performance when fixing the pre-trained model, while the green number denotes the best pre-trained model for each algorithm. Next we analyze the empirical result for the five datasets respectively.

4 Conclusion: Key Observations and Tips

Dataset	ERM DA	 Mixup	 GroupMix	GroupDRO	CORAL
Waterbirds	82.5	83.9	86.3	**88.0**	82.4
FMoW	36.6	37.0	**37.7**	36.9	37.1
Camelyon	91.1	90.6	90.7	**91.4**	90.3
iWildCam	35.4	36.0	36.0	18.0	**37.0**
DomainNet	**45.8**	44.7	44.2	37.8	45.7

Table 7: The performance of five learning algorithms on all datasets, averaged over pre-trained models.

Self-supervised or supervised pre-training? Fig. 1 shows the accuracy of seven pre-trained models, averaged over learning algorithms. For spurious correlation and subpopulation shift (Waterbirds and FMoW), self-supervised pre-training has a clear benefit over the supervised pre-training, when using the IN-1k. It indicates that the high-level features learned in supervised training are not robust to spurious correlation and subpopulation shift. On Camelyon, MAE is better than the remaining models including ViT and R50 trained on IN-21k, since the tissue images only contain low-level features. In contrast, the self-supervised model has not much benefit over supervised ones on iWildCam and DomainNet for ViT models. The OOD generalization in object recognition may need high-level features from the labels in pre-training. For ResNet, when the pre-training data is insufficient, MoCo-R50 is better than Sup. R50 on IN-1k.

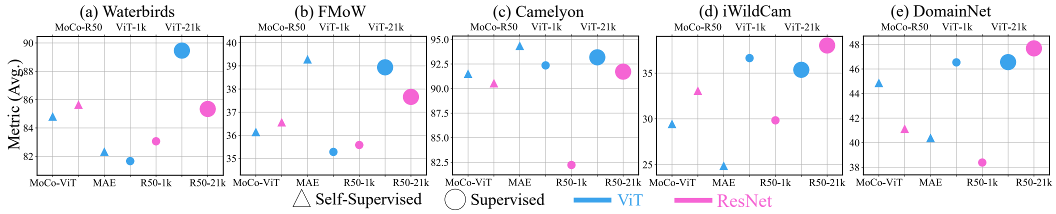

Figure 1: The performance of seven pre-trained models on five datasets, averaged over five algorithms. The triangle and circle denote self-supervised and supervised pre-trained model, while blue and pink denote ViT and R50. The large and small marker mean pre-training with IN-21k and IN-1k respectively.

			ERM			GroupDRO	CORAL
PT mode	Model	PT data	Data Aug.	Mixup	GroupMix		
MoCo	ViT	IN-1k	**47.5(0.1)**	46.7(0.1)	46.3(0.2)	37.6(0.0)	46.2(1.1)
MoCo	R50	IN-1k	**44.2(0.1)**	41.6(0.1)	41.3(0.2)	35.9(0.1)	42.6(0.3)
MAE	ViT	IN-1k	**43.6(0.2)**	41.6(0.5)	41.5(0.6)	31.7(0.1)	43.5(0.1)
Sup.	ViT	IN-1k	47.7(0.1)	47.3(0.1)	47.3(0.2)	42.4(0.0)	**48.0(0.0)**
	R50	IN-1k	40.9(0.1)	39.2(0.1)	37.3(0.2)	33.3(0.2)	**41.5(0.1)**
	ViT	IN-21k	47.4(0.2)	47.8(0.2)	47.6(0.3)	40.8(0.2)	**52.1(1.1)**
	R50	IN-21k	**49.8(0.1)**	48.5(0.1)	48.2(0.2)	43.1(0.0)	48.8(0.1)
Avg. Over Models			**45.8**	44.7	44.2	37.8	45.7

Table 6: Experimental Results on DomainNet. The six domains are set as held-out target domain respectively and the averaged accuracy of six experimental results is reported. On the object recognition data, supervised pre-training on IN-21k triumphs over other models.

ResNet or ViT? On Waterbirds, FMoW and Camelyon, ViT's achieve the best performance, while on iWildCam and DomainNet, the supervised pre-trained R50 on IN-21 achieves a better result than the ViT counterpart. On Waterbirds, ViT's benefit is quite substantial when pre-trained on IN-21k, indicating that with more fine-grained labels, the spurious correlation problem can be alleviated as a result of narrowed ambiguity. On OOD generalization datasets (Camelyon, iWildCam and DomainNet), an interesting phenomenon is that ViT is more data-efficient in supervised pre-training than R50: ViT outperforms R50 with a large margin using IN-1k while the benefit is washed out if using the larger IN-21k.

Larger pre-training dataset leads to stronger robustness? Comparing the performance of different pretraining sizes (corresponding to marker sizes in Fig. 1), the larger dataset IN-21k substantially improves the performance of IN-1k in most cases. Thus, it is validated that generalization under distribution shift in a downstream can be achieved by pre-training on a standard large-scale dataset.

Which learning algorithm is best? Tab. 7 shows the averaged accuracy of each algorithm over pre-trained models. GroupDRO is the best training method on average when evaluated on Waterbirds and Camelyon, where the data size is quite small and the difficulty of classification is not high (both are binary classification task). In contrast, on large-scale datasets such as iWildCam and DomainNet, GroupDRO is the worse one compared with others, indicating that GroupDRO is specialized for small-scale data. CORAL is quite competitive on iWildCam and DomainNet but not on small datasets such as Waterbirds. The possible reason is that CORAL relies on estimated means and variances of features, whose stability and accuracy will be affected if the data is scarce. ERM with GroupMix is a strong baseline on Waterbirds and FMoW, showing the importance of using group-information in spurious correlation with spurious correlation and subpopulation shift. On Camelyon, iWildCam and DomainNet, ERM with general DA is also quite competitive, ranking the first or second on the three datasets.

Tips. Finally, from these observations, we provide tips for practitioners for improving robustness to distribution shift:

1. For OOD generalization and spurious correlation in object recognition, use a supervised pre-trained model with as much pre-training data as possible. When the pre-training data is limited, self-supervised pre-trained models can be better.
2. If the target data is small, use algorithms respecting the group information such as GroupDRO and GroupMix. If the target data is large, CORAL and ERM with general DA are quite competitive.
3. For distribution shift in dense image classification, use a patch-based self-supervised pre-trained model, e.g., MAE.

Acknowledgement

This work was supported by Alibaba Group through Alibaba Research Intern Program, the Fundamental Research Funds for the Central University of China (DUT No. 82232031) and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11215820).

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[3] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[4] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

[5] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[6] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[7] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[8] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[9] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR, 2020.

[10] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

[11] Yo Joong Choe, Jiyeon Ham, and Kyubyong Park. An empirical study of invariant risk minimization. *arXiv preprint arXiv:2004.05007*, 2020.

[12] Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R Varshney. Empirical or invariant risk minimization? a sample complexity perspective. *arXiv preprint arXiv:2010.16412*, 2020.

[13] John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

[14] Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29, 2016.

[15] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Heterogeneous risk minimization. In *International Conference on Machine Learning*, pages 6804–6814. PMLR, 2021.

[16] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *Proceeding of the Thirty-ninth International Conference on Machine Learning*, 2022.

[17] Alexander Robey, George J. Pappas, and Hamed Hassani. Model-based domain generalization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20210–20229. Curran Associates, Inc., 2021.

[18] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7721–7735. PMLR, 18–24 Jul 2021.

[19] Chengqiu Dai, Fan Li, Xiyao Li, and Don Xie. Cadg: A model based on cross attention for domain generalization. *ArXiv*, abs/2203.17067, 2022.

[20] Papers with code: DomainNet Benchmark (domain generalization). https://paperswithcode.com/sota/domain-generalization-on-domainnet.

[21] WILDS Benchmark. https://wilds.stanford.edu/leaderboard/.

[22] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[24] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[25] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.

[26] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, jan 2016.

[27] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[28] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[29] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

[30] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*, 2022.

[31] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[32] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. In *International Conference on Learning Representations*, 2021.

[33] Yaodong Yu, Heinrich Jiang, Dara Bahri, Hossein Mobahi, Seungyeon Kim, Ankit Singh Rawat, Andreas Veit, and Yi Ma. An empirical study of pre-trained vision models on out-of-distribution generalization. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[34] Yuge Shi, Imant Daunhawer, Julia E Vogt, Philip HS Torr, and Amartya Sanyal. How robust are pre-trained models to distribution shift? *arXiv preprint arXiv:2206.08871*, 2022.

[35] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[36] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[37] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 2018.

[38] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[39] Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020.

[40] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[41] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[42] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[43] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[44] Chunting Zhou, Xuezhe Ma, Paul Michel, and Graham Neubig. Examining and combating spurious features under distribution shift. In *International Conference on Machine Learning*, pages 12857–12867. PMLR, 2021.

[45] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

[46] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

[47] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[48] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[49] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[50] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[52] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[53] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[54] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

[55] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[56] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624. Curran Associates, Inc., 2020.

[57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[58] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[59] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

A An Overview of Distribution Shift

We summarize two common sources of distribution shift in practice. The first source is the bias in data, as a result of real-world bias. Assuming the data distribution is $p(\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{y}$ is an attribute vector, the conditional distribution of one attribute given another $p(y^i|y^j)$ may be high, indicating a spurious correlation between y^i and y^j, e.g., the background and bird species in WILDS-Waterbirds. Some marginal distributions of $p(y^i)$ may be relatively small or large, leading to a subpopulation shift, e.g., the size of African images in WILDS-FMoW. The second source is the scarcity of data, i.e., the dataset cannot contain all possible images from the support of $p(\boldsymbol{x}, \boldsymbol{y})$ as a result of the high dimensionality of the continuous space. In other words, the training set contains data from a certain distribution $p_{tr}(\boldsymbol{x}|\boldsymbol{y})$ but the test set have a different distribution $p_{te}(\boldsymbol{x}|\boldsymbol{y}) \neq p_{tr}(\boldsymbol{x}|\boldsymbol{y})$, e.g., WILDS-iWildCam [3] and DomainNet [31]. Note that these two types of distribution shifts could be conceptually addressed by deliberately sampling a balanced dataset to avoid spurious correlation and subpopulation shift, and collecting all possible data from the distribution support. But in reality, we cannot obtain such a perfect dataset to train the model in many real-world applications. A pre-trained model on a balanced large-scale dataset is often used as the initialization when training on such downstream tasks, because the pre-trained model learns good representations and benefits the optimization. Our work mainly investigates the impact of pre-training on distribution shift, and aims to answer the following question: How does the pre-training, i.e., training on the standard large-scale dataset that is more carefully curated, affect the fine-tuning result on downstream datasets with distribution shift? To this end, we study different aspects of pre-training including training strategies, neural architectures and pre-training data sizes in our paper and summarize our findings in Section 4.

B Experimental Settings

In this section we discuss our experiment settings, including datasets, pre-trained models, and learning algorithms.

B.1 Datasets

To have a comprehensive evaluation on the effect of pre-training and data augmentation, we use five datasets that represent different types of distribution shifts. A general description of the five datasets follows. For more details about the datasets, see the original papers [3] and [31].

WILDS-Waterbirds [3, 5] is a bird classification task and has two kinds of birds in two types of environments. Specifically, bird images from CUB [40] are cropped and pasted to scene images from Places [41]. The training set of Waterbirds has 3,498 landbirds on land, 1,057 waterbirds on water, 184 landbirds on water and 56 waterbirds on land, inducing a spurious correlation in the data: the environment label (land/water) is correlated with the target label (waterbird/landbird). If the classical ERM is used, the environment features will be exploited and the performance of minor groups will be substantially worse than of major groups. Following the common practice [5, 42, 43, 44, 8], we use the worse-group (WG) accuracy as a metric for robustness to spurious correlation and report the average accuracy weighted by the *training* group size. As in WILDS benchmark [3], we use the input resolution of 224×224, the training epoch of 200 and batch size of 128 in this paper. We search the learning rate from $\{$0.000001,0.000002, 0.000004, 0.000008, 0.00001, 0.00002, 0.00004, 0.00008$\}$ and weight decay (WD) from $\{$0.0,1e-3,1e-2,1e-1,1.0$\}$, α of mixup from $\{$0.1,0.3,1.0$\}$. For GroupMix, we search the constant C from $\{$1.0,3.0,10.0,30.0$\}$. For GroupDRO, the constant C_{DRO} is searched from $\{$1.0,2.0,4.0,8.0,16.0$\}$ and the step size η_q is fixed as 0.01 for all datasets. For CORAL, the λ is searched from $\{$0.3,1.0,3.0$\}$. Thus, in the hyperparameter search stage, we train 960 models.

WILDS-FMoW [3, 38] is a satellite imagery data consisting of dense objects in the image. We use the satellite images to predict land usage as in [3, 45, 46]. The images are divided into five groups according to their regions (Asia, Europe, Africa, Americas and Oceania). Due to historical and economic reasons, developing regions have fewer satellite images than developed regions. So in the FMoW data, the Africa region (1,582 training images) has much fewer images than Europe (34,816 training images) or Americas (20,973 training images). With the unbalanced groups and the fact that image features often vary across different continents, training with classical ERM may have a lower accuracy in minor groups compared with major groups, and has the risk of learning biased models with discriminatory prediction. The input resolution is set as 224×224, the training epoch as 50 and batch size as 72. We fix the WD as 0.0 following existing practice and search the learning rate from $\{0.00001, 0.0001, 0.001\}$ and the α from $\{0.1,0.3,1.0\}$. We do not use the adjustment GroupDRO on this dataset and the following datasets. For GroupMix, we search C from $\{3.0,10.0,30.0,100.0\}$. The λ in CORAL is searched from $\{0.3,1.0,3.0\}$. We trained 60 models for hyperparameter search on FMoW.

WILDS-Camelyon [3, 37] contains tissue slide images from five hospitals, and the task is to predict the presence of tumor tissue. The ways to collect and process the slide images differ among hospitals, while it is often desired that the trained model on limited data from a few hospitals generalizes to unseen data from a novel environment. Three hospitals are used for training, one hospital is for validation and the other is for test. The input resolution is set as 96×96, the training epoch as 10 and batch size as 168. The WD and learning rate are searched from $\{0.000001, 0.000003, 0.00001, 0.00003, 0.0001, 0.0003, 0.001, 0.003, 0.01\}$ and $\{0.0,1e\text{-}5,1e\text{-}4,1e\text{-}3,1e\text{-}2\}$. The α is searched from $\{0.1,0.3,1.0\}$. For GroupMix, C is searched from $\{3.0,10.0,30.0,100.0\}$. For CORAL, the λ is searched from $\{0.3,1.0,3.0\}$. So we trained 900 models in total for Camelyon for hyperparameter search.

WILDS-iWildCam [3, 39] is collected by cameras in the wild, and the task is to predict wild animal species. The data are split into 323 groups according to the location of cameras since the environment has an impact on the image features. The trained model on certain groups is expected to generalize to OOD, i.e., images from unseen or new cameras. As in [3], we use the Macro F1 score as the metric of the performance on iWildCam. The input resolution is set as 448×448, the training epoch as 12 and batch size as 24. The learning rate is searched. The WD is fixed as 0.0, according to existing experiment in [3] and the learning rate is searched from $\{0.000001, 0.000002, 0.000004, 0.000008, 0.00001, 0.00002, 0.00004, 0.00008,0.0001\}$. The α in mixup is searched from $\{0.1,0.2,0.4,1.0\}$. For GroupMix, we search the constant C from $\{1.0,3.0,10.0,30.0\}$. For CORAL, the λ is searched from $\{0.3,1.0,3.0\}$. In the hyperparameter search stage, we train 225 models.

DomainNet [31] is often used to test domain generalization or adaptation, and consists of six domains that have the same categories but different image styles, e.g., painting, real and clipart. There are several other domain generalization datasets but we use the DomainNet because it is the largest dataset among its counterparts, e.g., PACS [47], VLCS [48], OfficeHome [49] and TerraIncognita [50]. We follow the common practice [4] to train a model on five domains and evaluate the performance on the held-out domain. Thus, we train six models since we repeat the process to test on each domain, and report the average results. Following [4], the input resolution is set as 224×224 and the training lasts for 5000 update steps. The batch size is 32 for each environment, resulting in a 32*5=160 samples for the whole batch. We search the learning rate from $\{0.00001, 0.0001, 0.001\}$, WD from $\{0.0,1e\text{-}5,1e\text{-}4\}$, α from $\{0.1,0.3,1.0\}$. The C in GroupMix is searched from $\{1.0,3.0,10.0\}$, the λ in CORAL is searched from $\{0.3,1.0,3.0\}$. This amounts to 810 trained models in this stage.

Finally, we use the PyTorch [51] code from WILDS [3] and DomainBed [4] benchmark. All of our experiment is run on a single Nvidia-V100 GPU with 32GB.

B.2 Pre-Trained Models

The benchmarks in WILDS and DomainNet use a pre-trained ResNet [1] on ImageNet-1k [23] as the initialization for training. Specifically, Waterbirds and iWildCam use ResNet50 [1], FMoW and Camelyon use DenseNet121 [52]. In a domain generalization benchmark [4], the pre-trained ResNet50 is the default model for DomainNet. In this work, the following seven pre-trained models are evaluated to study the impact of pre-training strategies, neural architectures and pre-training data size on the distribution shifts.

MoCo-R50 and MoCo-ViT-B/16 [35, 53] are self-supervised pre-trained models using contrastive learning to learn generic visual features. We use pre-trained ResNet50 (R50) and ViT-B/16 from [53].

MAE-ViT-B/16 [36] uses a masked image modeling task for self-supervised pre-training of the patch-based transformer architecture. We use the ViT-B/16 version of MAE.

Supervised ViT-B/16 and R50 that are trained on ImageNet-1k (IN-1k) and ImageNet-21k (IN-21k) are evaluated. There are 1,281,167 images and 1,000 classes in IN-1k, and 14,197,122 images and 21,841 classes in IN-21k. We use ViT-IN-1k and ViT-IN-21k model from [54]. The R50-IN-21k model is from [55]. Since all pre-trained ViT models have the same architecture, i.e., ViT-B/16, we use ViT for short in this paper. By comparing the performance of models pre-trained on 1k and 21k, we can investigate the impact of pre-training data scale on downstream distribution shifts.

B.3 Learning Algorithms

ERM is the classical objective function used in ML. We note that in stochastic gradient descent, the ERM is related to batch sampling. On Waterbirds and DomainNet, we follow the common practice [5, 4] to use the weighted sampling so that each batch has the same number of samples for all groups. On other datasets, the standard batch sampling is used.

Data augmentation. On Waterbirds, we use the standard data augmentation, i.e., random resizing and cropping, and random horizontal flipping. On Camelyon and FMoW, we use the 2-level of randaug [56]. On iWildCam, the 1-step randaug is used. On DomainNet, we use the standard data augmentation in Waterbirds, along with color jittering and random grayscale.

Mixup [57] optimizes a convex hull of training samples to improve the generalization, by augmenting the training data with random combination of input images and labels. Specifically, two samples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with labels y_1 and y_2 will generate a new training sample as follows

$$\lambda \sim Beta(\alpha, \beta), \quad \boldsymbol{x}' = \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2, \quad y' = \lambda y_1 + (1-\lambda) y_2, \tag{1}$$

where λ is sampled from a Beta distribution with parameters α and β.

GroupDRO [5]. GroupDRO takes $\{(\boldsymbol{x}_i, y_i, g_i)\}$ as the input, where the g_i is the group label for $\boldsymbol{x}_i$, and updates the network by minimizing a group-weighted loss. Specifically, the group weight of jth group is updated as

$$q_j(t+1) = q_j(t)\exp(\eta_q \frac{1}{N_j(t+1)} \sum_{\boldsymbol{x}_i \in g_j} l(\boldsymbol{x}_i, y_i)) = q_j(t)\exp(\eta_q l_j(t+1)), \tag{2}$$

where the η_q is the step size for group weight update, $N_j(t+1)$ is the number of samples from jth group in the current batch. For GroupDRO with adjustment, there is an extra term $C_{DRO}/\sqrt{n_j}$ added in $\eta_q l_j(t+1)$ of Equation 2. We use the shorthand notation $l_j(t+1)$ for the jth group's loss in the batch. Then the loss is weighted in a group-wise way,

$$\mathcal{L}(t+1) = q_j(t+1)\sum_{j=1}^{G} l_j(t+1). \tag{3}$$

GroupMix We add the group information in the mixup step,

$$\lambda \sim Beta(g_1^{(b)}\alpha, g_2^{(b)}\beta), \tag{4}$$

$$\boldsymbol{x}' = g_1^{(x)}\lambda \boldsymbol{x}_1 + g_2^{(x)}(1-\lambda)\boldsymbol{x}_2, \tag{5}$$

$$y' = g_1^{(l)}\lambda y_1 + g_2^{(l)}(1-\lambda) y_2. \tag{6}$$

The default GroupMix in our paper is to set $g_1^{(b)} = g_2^{(b)} = g_1^{(x)} = g_2^{(x)} = 1$ and $g_{1,2}^{(l)}$ is related to group size, i.e., the number of training samples in a group. Assume the number of training samples of jth group is n_j, the group weight is defined as the output of a softmax function $g_j = \text{softmax}(C/\sqrt{n_j})$ where the C is a hyperparameter. So the summation of group weights is one and smaller groups have larger weights. As in the main paper, we use the function $g(\cdot)$ for $\boldsymbol{x}$ to denote the group weight of $\boldsymbol{x}$.

We also investigate three variants of GroupMix, i.e.,

2) $g_1^{(b)} = \gamma g(\boldsymbol{x}_1)$ and $g_2^{(b)} = \gamma g(\boldsymbol{x}_2)$, keep other four weights to be 1. The $\gamma > 0$ is a constant hyperparameter to control the parameters in Beta distribution. To keep a stable sampling for two input images from the same group, we fix the α, β to be 0.1 if two images are from the same group. This variant uses the property of Beta distribution to give a larger λ for an image from minor groups.

3) $g_1^{(b)} = \gamma g(\boldsymbol{x}_1)$ and $g_2^{(b)} = \gamma g(\boldsymbol{x}_2)$, $g_1^{(x)} = g_2^{(x)} = 1.0$ and $g_1^{(l)} = g(\boldsymbol{x}_1), g_2^{(l)} = g(\boldsymbol{x}_2)$. Then normalize each sample's weight so that the summation of weight for a batch is 1. The difference between this method and 2) is that the label is multiplied by group weight and the coefficient/weight for losses is normalized.

4) $g_1^{(x)} = g_2^{(x)} = g_1^{(b)} = g_2^{(b)} = 1$, $g_1^{(l)} = g(\boldsymbol{x}_1), g_2^{(l)} = g(\boldsymbol{x}_2)$ and normalize the loss weights as in the third variant. This is the normalized version of the default GroupMix.

	Waterbirds		FMoW		Camelyon	iWildCam
Metric	WG Acc.	Avg. Acc.	WG Acc.	Avg. Acc.	OOD Acc.	OOD Macro F1
ERM	63.7(1.9)	**97.0(0.2)**	34.8(1.90)	**55.6(0.23)**	70.8(7.2)	32.0(1.5)
GroupDRO	**91.4(1.1)**	93.5(0.3)	30.8(0.18)	52.1(0.50)	68.4(7.3)	23.9(2.1)
IRM	67.4(5.2)	73.4(9.7)	30.0(1.37)	50.8(0.13)	64.2(8.1)	15.1(4.9)
CORAL	79.4(1.9)	94.1(0.9)	31.7(1.24)	50.5(0.36)	59.5(7.7)	**32.8(0.1)**
ERM+data aug	80.7(1.1)	93.6(1.4)	34.8(1.48)	55.4(0.52)	82.0(7.4)	32.2(1.2)
ERM+Mixup	82.6(2.5)	93.7(1.3)	36.8(0.93)	55.0(0.73)	91.8(0.7)	**32.8(1.4)**
ERM+GroupMix	85.0(2.3)	89.6(2.6)	**39.0(1.22)**	54.8(0.92)	**92.7(0.2)**	31.9(0.9)

Table 8: WILDS benchmarks with data augmentation, mixup and GroupMix, using the default pre-trained model in each dataset. With mixup or GroupMix, the ERM outperforms GroupDRO, IRM and CORAL on FMoW, Camelyon and iWildCam.

PT Model	Metric	GroupMix				GroupDRO
		V1	V2	V3	V4	
Sup.,ViT,IN-21k	WG Acc.	91.5(0.8)	92.0(1.1)	**92.6(0.8)**	91.9(0.9)	**92.6(0.5)**
	Avg. Acc.	93.6(1.0)	95.6(0.6)	94.1(0.7)	96.0(0.5)	93.2(0.5)

Table 9: Worse-group (Row 3) and average (Row 4) test accuracy of different versions of GroupMix on Waterbirds, compared with GroupDRO. GroupMix-v3 achieves the same worse-group accuracy as GroupDRO, demonstrating the potential of GroupMix. Also note that the average accuracy of GroupMix is generally higher than that of GroupDRO.

CORAL [6] aligns the feature means and covariances of different groups by minimizing the squared L2/Frobenius norm of the difference in means/covariances, i.e.,

$$\lambda(\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k\|_2^2 + \|\boldsymbol{\Sigma}_j - \boldsymbol{\Sigma}_k\|_F^2), \tag{7}$$

where the j and k denote two groups and the mean and covariance are computed from the features of two groups, i.e., the last backbone layer's output.

B.4 Training Settings

We use the official training/validation/testing split for WILDS datasets. For DomainNet, if a domain is used for training, then the data is randomly split into 80% and 20% as training and validation set following [4]. As the de facto setting in WILDS and DomainBed benchmark, the training starts with a pre-trained model and optimizes all parameters with the training data. In other words, we study the performance of fine-tuning all parameters of a pre-trained model on distribution-shift datasets. The training epochs/steps are set as the same in the WILDS [3] or DomainBed [4] benchmark. On WILDS datasets, we use the official (out-of-distribution) validation set to do the model selection. On DomainNet, we use the training domain validation set in the model selection, as suggested by [4]. In Table 8, the optimizer is the same as the benchmark. In all other experiments, AdamW [58] is used as the optimizer. We search the hyperparameters for each setting by the grid search, where the parameter space is determined by either existing training practice or our pilot study on those datasets and models. For the search space for each dataset, please see the appendix. In total, we trained about 3000 models during the hyperparameter searching. Following the common practice [3, 4], on WILDS datasets, we report the mean and standard deviation of 5 trials; on DomainNet, we report the result of 3 trials. The model selection (early-stopping) uses the default validation ood set, i.e., choose the model with the best validation accuracy. Different from the original WILDS benchmark, the hyperparameters are chosen based on the accuracy of the test ood set, instead of the validation ood set.

C More Experimental Result

Tab. 8 shows the performance of ERM with DA, mixup and GroupMix, compared on several WILDS baselines using the pre-trained models. On Camelyon, we use a pre-trained DenseNet121, which is different from the default setting of WILDS benchmark. Except for Waterbirds, ERM with general DA achieves the best result. On Waterbirds, GroupMix substantially improves ERM with other types of data augmentation. For the result of ERM on DomainNet, [4] draws a similar conclusion that ERM with data augmentation is a quite strong baseline on all popular domain generalization datasets.

	Waterbirds		FMoW	
Metric	WG Acc.	Avg. Acc.	WG Acc.	Avg. Acc.
ERM	63.7(1.9)	**97.0(0.2)**	34.8 (1.9)	55.6 (0.2)
ERM+data aug	80.7(1.1)	93.6(1.4)	34.8(1.5)	55.4 (0.5)
GroupDRO	91.4(1.1)	93.5(0.3)	30.8(0.2)	52.1 (0.5)
IRM	67.4(5.2)	73.4(9.7)	30.0(1.4)	50.8 (0.1)
CORAL	79.4(1.9)	94.1(0.9)	31.7(1.2)	50.5 (0.4)
ERM (MoCo, ViT-B, IN-1k)	84.7(1.3)	90.1(1.94)	36.2(1.4)	55.8(0.5)
ERM (MoCo, R50, IN-1k)	82.5(0.8)	92.7(0.9)	36.4(2.3)	53.8(1.0)
ERM (MAE, ViT-B, IN-1k)	81.7(2.0)	92.8(2.1)	39.1(0.5)	58.1(0.4)
ERM (Sup., ViT-B, IN-1k)	76.3(2.4)	94.7(0.6)	34.5(0.8)	56.4(0.4)
ERM (Sup., R50, IN-1k)	80.7(1.1)	93.6(1.4)	34.9(1.9)	52.3(0.7)
ERM (Sup., ViT-B, IN-21k)	88.5(0.6)	96.2(0.7)	37.9(1.3)	59.2(0.8)
ERM (Sup., R50, IN-21k)	82.9(1.4)	96.0(0.5)	37.2(2.9)	55.1(1.3)
GroupDRO (MoCo, ViT-B, IN-1k)	86.7(0.8)	90.1(0.9)	35.2(1.6)	55.5(1.0)
GroupDRO (MoCo, R50, IN-1k)	88.1(0.9)	90.6(0.6)	37.2(1.1)	54.5(0.7)
GroupDRO (MAE, ViT-B, IN-1k)	87.4(1.0)	88.6(0.8)	37.9(1.4)	57.8(0.6)
GroupDRO (Sup., ViT-B, IN-1k)	87.0(1.1)	88.8(1.4)	35.8(1.8)	56.5(0.4)
GroupDRO (Sup., R50, IN-1k)	87.6(0.4)	88.5(0.7)	36.1(1.6)	52.0(0.7)
GroupDRO (Sup., ViT-B, IN-21k)	**92.6(0.5)**	93.2(0.5)	39.0(1.0)	59.2(0.5)
GroupDRO (Sup., R50, IN-21k)	86.6(1.1)	93.5(0.3)	37.1(1.9)	55.9(0.7)
CORAL (MoCo, ViT-B, IN-1k)	83.3(1.9)	93.1(1.2)	35.8(1.7)	53.5(1.7)
CORAL (MoCo, R50, IN-1k)	83.2(1.8)	92.6(1.7)	36.1(0.7)	53.3(0.7)
CORAL (MAE, ViT-B, IN-1k)	80.0(2.3)	93.7(0.6)	40.4(1.0)	54.8(0.8)
CORAL (Sup., ViT-B, IN-1k)	81.4(1.7)	94.2(1.3)	35.9(1.2)	53.1(1.5)
CORAL (Sup., R50, IN-1k)	79.4(1.9)	94.1(0.9)	34.2(1.2)	52.0(1.3)
CORAL (Sup., ViT-B, IN-21k)	86.5(1.3)	95.7(0.6)	**40.7(1.0)**	57.4(2.1)
CORAL (Sup., R50, IN-21k)	83.2(1.2)	96.1(0.5)	36.6(0.8)	54.6(1.4)
Mixup (MoCo, ViT-B, IN-1k)	83.5(2.7)	92.7(1.9)	37.0(1.7)	56.4(0.3)
Mixup (MoCo, R50, IN-1k)	86.7(0.7)	93.4(0.7)	36.0(1.2)	56.7(0.2)
Mixup (MAE, ViT-B, IN-1k)	80.3(1.5)	94.1(1.1)	38.8(1.1)	58.4(0.7)
Mixup (Sup., ViT-B, IN-1k)	79.5(4.5)	94.5(1.8)	34.4(0.3)	56.5(0.5)
Mixup (Sup., R50, IN-1k)	82.6(2.5)	93.7(1.3)	35.9(1.8)	54.6(0.7)
Mixup (Sup., ViT-B, IN-21k)	88.2(1.2)	96.9(1.3)	38.5(0.9)	60.0(0.2)
Mixup (Sup., R50, IN-21k)	86.5(1.3)	95.3(0.5)	38.4(0.6)	57.6(0.7)
GroupMix (MoCo, ViT-B, IN-1k)	85.8(1.4)	89.6(1.2)	36.5(1.5)	56.1(0.6)
GroupMix (MoCo, R50, IN-1k)	87.7(0.7)	91.0(0.7)	37.1(1.4)	55.2(1.3)
GroupMix (MAE, ViT-B, IN-1k)	82.2(2.5)	92.4(1.6)	40.2(1.6)	57.1(1.0)
GroupMix (Sup., ViT-B, IN-1k)	84.1(1.9)	90.8(1.9)	35.8(0.7)	54.4(0.8)
GroupMix (Sup., R50, IN-1k)	85.0(2.3)	89.6(2.6)	36.8(1.3)	53.5(1.7)
GroupMix (Sup., ViT-B, IN-21k)	91.5(0.8)	93.6(1.0)	38.6(1.5)	**60.1(0.2)**
GroupMix (Sup., R50, IN-21k)	87.5(1.6)	93.5(0.6)	39.0(0.8)	58.1(0.5)

Table 10: The full result of Waterbirds and FMoW with worse-group accuracy and averaged accuracy.

C.1 Variants of GroupMix

We evaluate four variants of GroupMix on Waterbirds. In addition to original V1, we consider the following 3 variants: 2) the weight $g_1^{(b)}, g_2^{(b)}$ is proportional to $g(\boldsymbol{x}_1), g(\boldsymbol{x}_2)$, which gives larger λ to a sample from minor groups, and keep other weights 1; 3) $(g_1^{(b)}, g_2^{(b)}) \propto (g(\boldsymbol{x}_1), g(\boldsymbol{x}_2))$, $g_1^{(x)} = g_2^{(x)} = 1.0, g_1^{(l)} = g(\boldsymbol{x}_1), g_2^{(l)} = g(\boldsymbol{x}_2)$ and normalize the loss weight so that the weight sums to 1 as in ERM; 4) $g_1^{(b)} = g_2^{(b)} = g_1^{(x)} = g_2^{(x)} = 1.0$, $g_1^{(l)} = g(\boldsymbol{x}_1), g_2^{(l)} = g(\boldsymbol{x}_2)$ and normalize the weight as in 3). Tab. 9 shows the result of four variants, where the GroupMix-V3 is the best one in terms of worse-case accuracy and matches the performance of GroupDRO. It indicates that the group-condition sampling can further improve the performance of GroupMix. Moreover, all versions of GroupMix have a higher average accuracy than GroupDRO, because mixup alleviates the overfitting in training.

A) Model selection: Out-of-distribution validation set

	ERM						GroupDRO		CORAL	
PT Model	Data Aug.		Mixup		GroupMix					
MoCo-ViT-IN-1k	34.0(0.5)	51.1(1.0)	31.7(2.1)	49.3(1.1)	32.8(0.7)	48.7(1.7)	13.5(1.0)	22.6(1.2)	35.2(0.6)	47.5(2.1)
MoCo-R50-IN-1k	34.5(1.5)	51.7(1.2)	36.8(1.6)	51.2(1.4)	36.0(1.1)	50.2(2.0)	20.8(0.8)	30.2(1.0)	37.2(1.1)	51.1(1.6)
MAE-ViT-IN-1k	28.4(2.0)	44.3(1.7)	27.3(2.2)	44.6(1.9)	28.0(2.9)	44.5(2.0)	9.2(0.9)	14.7(1.1)	31.4(2.6)	42.1(1.2)
Sup-ViT-IN-1k	38.9(0.7)	53.6(1.3)	41.0(0.7)	54.7(1.2)	40.9(1.2)	54.6(1.1)	20.5(0.5)	30.2(1.1)	41.9(0.9)	51.8(1.5)
Sup-R50-IN-1k	32.2(1.2)	47.0(1.4)	32.8(1.4)	49.6(0.6)	31.9(0.9)	48.4(1.2)	18.9(0.8)	28.9(0.7)	33.4(0.5)	48.1(1.2)
Sup-ViT-IN-21k	39.0(2.7)	55.4(2.4)	**41.3(2.4)**	**55.7(2.1)**	41.1(1.4)	**57.2(0.4)**	19.0(1.2)	28.7(1.5)	36.4(1.6)	47.3(1.8)
Sup-R50-IN-21k	**40.9(1.3)**	**55.5(0.9)**	40.8(0.4)	55.1(1.0)	**41.2(1.3)**	55.0(1.0)	**24.0(0.8)**	**33.4(0.7)**	**43.2(0.9)**	**52.1(2.4)**

B) Model selection: In-distribution (ID) validation set

	ERM						GroupDRO		CORAL	
PT Model	Data Aug.		Mixup		GroupMix					
MoCo-ViT-IN-1k	32.5(0.8)	49.8(0.6)	30.6(1.6)	47.8(1.7)	33.6(1.5)	48.8(1.5)	13.4(0.5)	21.9(1.7)	36.2(0.7)	48.9(2.0)
MoCo-R50-IN-1k	34.9(1.6)	51.7(1.1)	36.5(1.4)	50.6(1.3)	36.1(1.1)	49.0(1.5)	20.8(0.8)	30.7(0.8)	37.9(1.2)	50.6(1.6)
MAE-ViT-IN-1k	28.6(1.6)	43.9(1.4)	27.4(1.8)	44.7(1.9)	28.1(3.0)	43.9(2.7)	9.6(1.3)	14.8(1.5)	31.8(1.9)	42.3(1.8)
Sup-ViT-IN-1k	39.1(0.9)	54.0(1.0)	40.3(1.5)	54.5(2.0)	40.2(1.4)	53.2(0.8)	20.3(0.4)	30.0(0.8)	42.2(0.9)	51.5(1.2)
Sup-R50-IN-1k	31.1(0.9)	45.6(0.4)	32.6(1.2)	49.9(0.9)	31.8(0.7)	47.7(1.0)	19.3(0.6)	29.8(1.3)	34.4(0.9)	47.6(1.4)
Sup-ViT-IN-21k	38.4(2.9)	54.9(2.0)	**40.7(1.9)**	**55.3(2.1)**	**41.6(2.2)**	**56.1(1.2)**	18.7(1.5)	28.8(1.5)	36.6(0.8)	47.5(1.2)
Sup-R50-IN-21k	**40.0(0.6)**	**55.1(1.1)**	40.4(0.8)	54.8(1.6)	40.4(2.0)	54.8(1.6)	**23.9(0.9)**	**33.5(0.8)**	**43.2(0.8)**	**52.8(1.7)**

Table 11: WILDS-iWildCam result of OOD and ID validation set model selection. The left and right cell of each learning algorithm denote OOD and ID Macro F1 score. The best performance in each column is highlighted.

	clipart	infograph	painting	quickdraw	real	sketch	Avg
ERM (MoCo, ViT-B, IN-1k)	66.8 ± 0.2	24.2 ± 0.3	54.5 ± 0.1	17.3 ± 0.3	65.7 ± 0.1	56.4 ± 0.1	47.5 ± 0.1
ERM (MoCo, R50, IN-1k)	62.5 ± 0.1	22.0 ± 0.2	50.5 ± 0.1	14.5 ± 0.2	62.4 ± 0.1	53.3 ± 0.1	44.2 ± 0.1
ERM (MAE, ViT-B, IN-1k)	61.8 ± 0.1	21.5 ± 0.1	49.4 ± 0.4	17.4 ± 0.4	59.6 ± 0.4	51.7 ± 0.3	43.6 ± 0.2
ERM (Supervised, ViT-B, IN-1k)	67.5 ± 0.2	23.6 ± 0.2	54.1 ± 0.3	17.6 ± 0.2	68.7 ± 0.1	54.7 ± 0.2	47.7 ± 0.1
ERM (Supervised, R50, IN-1k)	58.1 ± 0.3	18.8 ± 0.3	46.7 ± 0.3	12.2 ± 0.4	59.6 ± 0.1	49.8 ± 0.4	40.9 ± 0.1
ERM (Supervised, ViT-B, IN-21k)	68.1 ± 0.3	23.1 ± 0.2	54.1 ± 0.1	18.0 ± 0.5	65.6 ± 0.3	55.7 ± 0.2	47.4 ± 0.2
ERM (Supervised, R50, IN-21k)	67.7 ± 0.1	**26.5 ± 0.2**	**57.1 ± 0.1**	16.2 ± 0.1	**73.2 ± 0.1**	58.2 ± 0.2	49.8 ± 0.1
GroupDRO (MoCo, ViT-B, IN-1k)	50.8 ± 0.3	22.0 ± 0.1	43.4 ± 0.2	12.7 ± 0.1	53.3 ± 0.1	43.6 ± 0.1	37.6 ± 0.0
GroupDRO (MoCo, R50, IN-1k)	50.7 ± 0.4	18.5 ± 0.2	39.5 ± 0.2	10.0 ± 0.2	53.2 ± 0.3	43.4 ± 0.3	35.9 ± 0.1
GroupDRO (MAE, ViT-B, IN-1k)	44.5 ± 0.5	19.5 ± 0.4	32.8 ± 0.2	11.1 ± 0.4	45.6 ± 0.5	36.9 ± 0.4	31.7 ± 0.1
GroupDRO (Sup., ViT-B, IN-1k)	60.6 ± 0.2	20.9 ± 0.4	46.8 ± 0.4	13.8 ± 0.4	64.6 ± 0.1	47.6 ± 0.2	42.4 ± 0.0
GroupDRO (Sup., R50, IN-1k)	47.2 ± 0.5	17.5 ± 0.4	33.8 ± 0.5	9.3 ± 0.3	51.6 ± 0.4	40.1 ± 0.6	33.3 ± 0.2
GroupDRO (Sup., ViT-B, IN-21k)	58.8 ± 0.2	20.5 ± 0.7	44.9 ± 0.4	15.4 ± 0.5	59.6 ± 0.1	45.7 ± 0.5	40.8 ± 0.2
GroupDRO (Sup., R50, IN-21k)	59.1 ± 0.2	22.3 ± 0.1	49.6 ± 0.1	11.0 ± 0.1	66.4 ± 0.1	50.2 ± 0.2	43.1 ± 0.0
CORAL (MoCo, ViT-B, IN-1k)	66.8 ± 0.1	16.1 ± 6.5	54.9 ± 0.1	17.9 ± 0.3	65.5 ± 0.1	56.2 ± 0.6	46.2 ± 1.1
CORAL (MoCo, R50, IN-1k)	61.4 ± 0.6	20.6 ± 0.2	47.4 ± 0.2	13.8 ± 0.4	59.6 ± 0.4	52.5 ± 0.4	42.6 ± 0.3
CORAL (MAE, ViT-B, IN-1k)	61.9 ± 0.2	21.6 ± 0.1	49.4 ± 0.2	16.8 ± 0.4	59.7 ± 0.1	51.7 ± 0.5	43.5 ± 0.1
CORAL (Sup., ViT-B, IN-1k)	67.8 ± 0.2	23.9 ± 0.2	55.1 ± 0.3	17.5 ± 0.1	68.6 ± 0.2	55.2 ± 0.3	48.0 ± 0.0
CORAL (Sup., R50, IN-1k)	59.2 ± 0.1	19.7 ± 0.2	46.6 ± 0.3	13.4 ± 0.4	59.8 ± 0.2	50.1 ± 0.6	41.5 ± 0.1
CORAL (Sup., ViT-B, IN-21k)	**69.5 ± 0.1**	24.8 ± 0.1	56.5 ± 0.1	**19.9 ± 0.5**	67.2 ± 0.1	57.5 ± 0.4	**52.1 ± 1.1**
CORAL (Sup., R50, IN-21k)	67.2 ± 0.4	25.2 ± 0.2	55.7 ± 0.1	14.9 ± 0.3	71.5 ± 0.2	**58.6 ± 0.2**	48.8 ± 0.1
Mixup (MoCo, ViT-B, IN-1k)	65.0 ± 0.3	23.4 ± 0.1	54.3 ± 0.2	18.1 ± 0.5	63.7 ± 0.2	55.4 ± 0.3	46.7 ± 0.1
Mixup (MoCo, R50, IN-1k)	58.7 ± 0.5	19.0 ± 0.2	49.0 ± 0.4	13.2 ± 0.2	58.8 ± 0.2	51.2 ± 0.4	41.6 ± 0.1
Mixup (MAE, ViT-B, IN-1k)	59.7 ± 0.1	20.1 ± 0.7	47.6 ± 0.4	17.0 ± 0.1	55.0 ± 1.6	50.2 ± 0.2	41.6 ± 0.5
Mixup (Sup., ViT-B, IN-1k)	66.9 ± 0.2	23.4 ± 0.3	54.0 ± 0.4	17.1 ± 0.2	67.8 ± 0.2	54.4 ± 0.3	47.3 ± 0.1
Mixup (Sup., R50, IN-1k)	55.7 ± 0.3	18.5 ± 0.5	44.3 ± 0.5	12.5 ± 0.4	55.8 ± 0.3	48.2 ± 0.5	39.2 ± 0.1
Mixup (Sup., ViT-B, IN-21k)	68.4 ± 0.3	23.8 ± 0.6	54.4 ± 0.7	19.8 ± 0.4	65.1 ± 0.2	55.1 ± 0.6	47.8 ± 0.2
Mixup (Sup., R50, IN-21k)	65.7 ± 0.2	24.3 ± 0.2	57.0 ± 0.5	14.9 ± 0.1	70.9 ± 0.2	58.3 ± 0.2	48.5 ± 0.1
GroupMix (MoCo, ViT-B, IN-1k)	64.0 ± 0.5	23.6 ± 0.1	54.3 ± 0.4	17.5 ± 0.3	62.9 ± 0.6	55.1 ± 0.3	46.3 ± 0.2
GroupMix (MoCo, R50, IN-1k)	58.8 ± 0.2	19.5 ± 0.3	47.6 ± 0.8	12.9 ± 0.1	57.7 ± 0.5	51.3 ± 0.4	41.3 ± 0.2
GroupMix (MAE, ViT-B, IN-1k)	58.6 ± 0.9	20.7 ± 0.4	45.8 ± 2.5	17.2 ± 0.3	56.7 ± 0.1	50.2 ± 0.2	41.5 ± 0.6
GroupMix (Sup., ViT-B, IN-1k)	66.8 ± 0.3	23.4 ± 0.5	54.4 ± 0.9	17.3 ± 0.1	67.5 ± 0.1	54.3 ± 0.4	47.3 ± 0.2
GroupMix (Sup., R50, IN-1k)	52.9 ± 0.2	17.2 ± 0.3	43.3 ± 0.8	11.8 ± 0.4	51.5 ± 0.3	47.3 ± 0.2	37.3 ± 0.2
GroupMix (Sup., ViT-B, IN-21k)	67.8 ± 0.6	23.7 ± 0.5	54.5 ± 0.5	19.4 ± 0.5	65.4 ± 0.2	55.0 ± 0.6	47.6 ± 0.3
GroupMix (Sup., R50, IN-21k)	65.6 ± 0.1	24.3 ± 0.4	55.9 ± 0.7	14.3 ± 0.1	70.8 ± 0.3	58.2 ± 0.2	48.2 ± 0.2

Table 12: The full experimental result on DomainNet.

D The full experimental results

Tab. 10 shows the full result of Waterbirds and FMoW in our paper, including the averaged accuracy. On Waterbirds, it is worth noting that the averaged accuracy of GroupMix is generally higher than that of GroupDRO. On FMoW, even though the worse-group accuracy of CORAL is higher than GroupMix, the average accuracy of GroupMix is significantly higher than CORAL's. In conclusion, ERM with GroupMix is quite competitive in both worse-group accuracy and averaged performance.

Tab. 11 shows the result of WILDS with ID and OOD Macro F1. To compare the difference between ID and OOD validation set based model selection, we report the result using ID validation set in the model selection in Tab. 11.B. We observe that CORAL has the drawback of sacrificing the ID accuracy to achieve a high OOD accuracy, whil GroupMix does not reduce the ID accuracy compared with DA and Mixup. Thus, the ERM with data augmentation which uses group information is a strong baseline in both ID and OOD distribution. The table also shows that the two model selection methods do not differ at a significant level. For example, the top-1 models in Tab. 11.A and B overlap

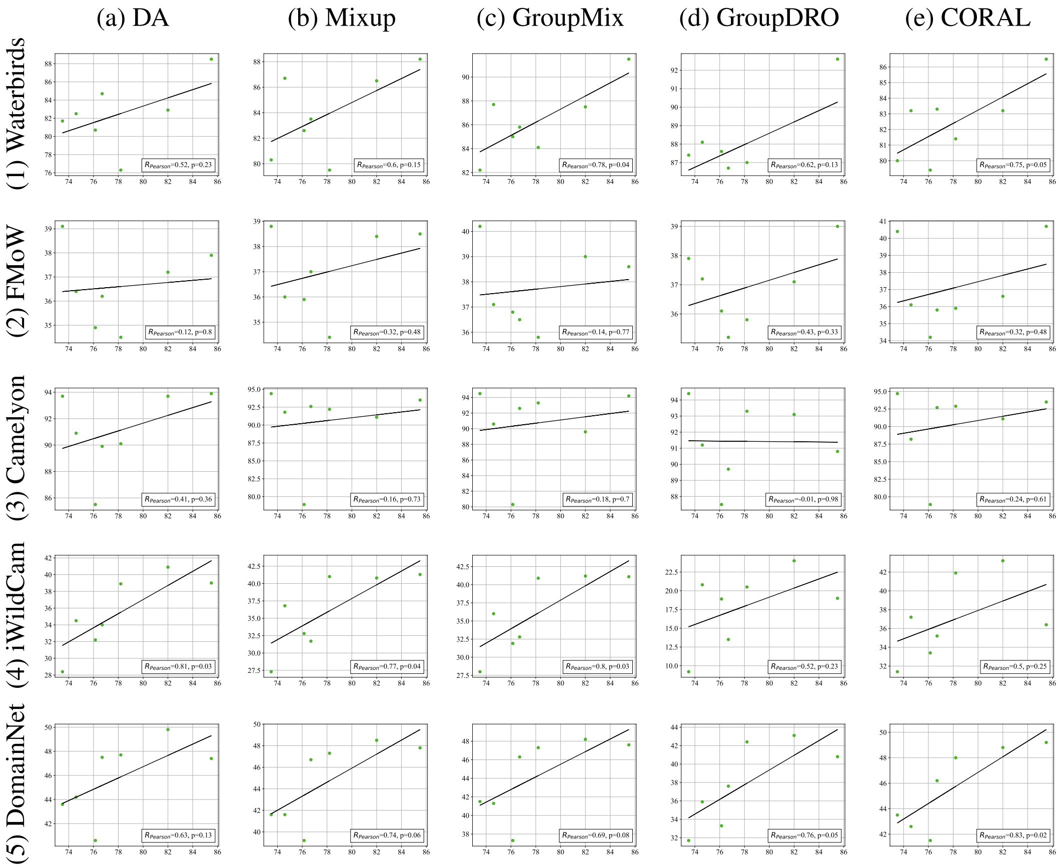

Figure 2: The correlation analysis between ImageNet accuracy of a pre-trained model and its downstream performance under distribution shift. The x-axis is IN accuracy and the y-axis is the performance of distribution shift in each dataset. The Pearson's R and p value of the linear regression are shown in each figure.

in most learning algorithms. So the ID validation set based model selection will not change our general conclusion in the main paper.

Tab. 12 reports the result of 6 domain generalization accuracy for each experimental setting. It is interesting that on the two most challenging domains, i.e., infograph and quickdraw, ViT and R50 supervised pre-trained on IN-21k have quite different performance. Sup-ViT-IN21k is better at generalizing to quickdraw, the most difficult domain, while Sup-R50-IN21k is better at infograph. This phenomenon indicates that the performance in domain generalization can be further improved by using an ensemble of different neural architectures.

Fig.2 shows the result of linear regression between ImageNet (IN) [23] test accuracy of a pre-trained model and its performance in a target task with distribution shift. For self-supervised models, we use the linear probing result as the IN accuracy as reported in their paper [36, 59, 35]. Fig.2 shows all the regression analyses of learning algorithms and datasets. For Waterbirds, the correlation between IN accuracy and downstream performance is statistically significant for GroupMix and CORAL. On iWildCam and DomainNet, the correlation is more obvious than on Waterbirds. However, on FMoW and Camelyon, there is no significant correlation between the two performances. This phenemenon further validates our hypothesis that for object recognition task, increasing the performance on the standard dataset (IN) is helpful for downstream tasks under distribution shift. But the benefit of IN performance is no longer valid if the downstream task has quite different visual features such as dense images in FMoW and Camelyon.